

WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŹDŹEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

SPRAWOZDANIE

KAMILA HANUSZ
GEOINFORMATYKA ROK II

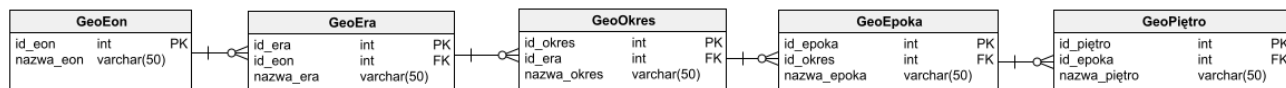
Chronostratygrafia tabela

Tylko jednostki rekomendowane przez ICS (bez prekambriu)

Enotem	Eratem	System	Oddział	Piętro	Wiek (mln lat)
fanerozoik	kenozoik	czwartorzęd	holocen		0,0117 ± 0,0001
			plejstocen	plejstocen górny	0,126
				jon	0,781
				kalabr	1,806
				gelas	2,6
		neogen	pliocen	piacent	3,6
				zankl	5,333
			miocen	messyn	7,246
				torton	11,63
				serrawal	13,82
				lang	15,97
				burdygał	20,44
				akwitan	23,03
		oleogen	oligocen	szat	28,1
				rupel	33,9
			eocen	priabon	37,8
				barton	41,2
				lutet	47,8
				lutet	47,8
		paleogen	eocen	priabon	37,8
				barton	41,2
				lutet	47,8
				iprez	56
			paleocen	tanet	59,2
				zeland	61,6
				dan	66
				dan	66
	mezozoik	kreda	kreda górna	mastrycht	72,1 ± 0,2
				kampan	83,6 ± 0,2
				santon	86,3 ± 0,5
				koniak	89,8 ± 0,3
				turon	93,9
				cenoman	100,5
			kreda dolna	alb	~ 113
				apt	~ 125
				barrem	~ 129,4
				hoteryw	~ 132,9
				walanżyn	~ 139,8
				berias	~ 145
		jura	jura górna	tyton	150,8 ± 4
				kimeryd	155,6
				oksford	161,2 ± 4
			jura środkowa	kelowej	164,7 ± 4
				baton	167,7 ± 3,5
				bajos	171,6 ± 3
				aalen	174,1 ± 1
			jura dolna	toark	183 ± 1,5
				pliensbach	189,6 ± 1,5
				synemur	199,53 ± 0,24
	trias	trias	trias górny	hetang	201,3 ± 0,2
				retyk	203,6 ± 1,5
				noryk	216,5 ± 2
				karnik	~ 237
			trias środkowy	ladyn	237 ± 2
				anizyk	~ 247,2
			trias dolny	olenek	249,7 ± 0,7
				ind	...

paleozoik	perm	loping	czangszing	🔑	253,8 ± 0,7
			wucziaping	🔑	260,4 ± 0,7
		gwadalup	kapitan	🔑	265,8 ± 0,7
			word	🔑	268 ± 0,7
			road	🔑	270,6 ± 0,7
		cisural	kungur		275,6 ± 0,7
			artyński		290,1 ± 0,2
			sakmar		285
			aselsk	🔑	299 ± 0,8
		karbon	pensylwan		318,1 ± 1,3
			missisip		359,2 ± 2,5
	dewon	dewon górny	famen	🔑	374,5 ± 2,6
			fran	🔑	385,3 ± 2,6
		dewon środkowy	żywet	🔑	391,8 ± 2,7
			eifel	🔑	397,5 ± 2,7
		dewon dolny	ems	🔑	407 ± 2,8
			prag	🔑	411,2 ± 2,8
			lochkow	🔑	416 ± 2,8
	sylur	przydol		🔑	418,7 ± 2,7
		ludlow	ludford	🔑	421,3 ± 2,6
			gorst	🔑	422,9 ± 2,5
		wenlok	homer	🔑	426,2 ± 2,4
			sheinwood	🔑	428,2 ± 2,3
		landower	telych	🔑	436 ± 1,9
			aeron	🔑	440,8 ± 1,2
			rhuddan	🔑	443,7 ± 1,5
	ordowik	ordowik górny	hirnant	🔑	445,6
			kat	🔑	455,8
			sandb	🔑	460,9
		ordowik środkowy	darriwil	🔑	468,1
			daping	🔑	471,8
		ordowik dolny	flo	🔑	478,6
			tremadok	🔑	488,3
	kambr	furong	piętro 10 [kambr]		~ 489,5
			jiangshan		~ 494
			paib	🔑	499
		oddział 3 [kambr]	gužang	🔑	503
			drum	🔑	506,5
		oddział 2 [kambr]	piętro 5 [kambr]		510
			piętro 4 [kambr]		515
			piętro 3 [kambr]		521
		terenew	piętro 2 [kambr]		528
			fortun	🔑	542
proterozoik	neoproterozoik	ediakar		🔑	635
		kriogen			850
		ton			1000
	mezoproterozoik	sten			1200
		ectas			1400
		kalym			1600
	paleoproterozoik	stater			1800
		orosir			2050
		riak			2300
		sider			2500
archaik	neoarchaik				2800
	mezoarchaik				3200
	paleoarchaik				3600
	eoarchaik				4600
hadeik					~ 4600

Znormalizowany schemat tabeli geochronologicznej



Konfiguracja sprzętowa i programowa

Testy wykonano na komputerze o następujących parametrach:

- ❖ CPU: AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
- ❖ RAM: 16.0 GB
- ❖ SSD: Samsung PM991 512GB
- ❖ S.O.: Windows 11

Jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

- ❖ SQL Server Management Studio 15.0.2000.5,
- ❖ PostgreSQL, wersja 14.2

Testy wykonywano dziesięciokrotnie dla każdego systemu zarządzania bazą danych.

SQL Server:

```
--Utworzenie bazy danych o nazwie "Geochronologia"
CREATE DATABASE Geochronologia;

--Stworzenie sześciu tabel oraz dodanie do nich kluczy obcych
CREATE TABLE GeoEon (
    id_eon INTEGER NOT NULL PRIMARY KEY,
    nazwa_eon VARCHAR(50) NOT NULL
);

CREATE TABLE GeoEra (
    id_era INTEGER NOT NULL PRIMARY KEY,
    id_eon INTEGER FOREIGN KEY REFERENCES GeoEon(id_eon),
    nazwa_era VARCHAR(50) NOT NULL
);

CREATE TABLE GeoOkres (
    id_okres INTEGER NOT NULL PRIMARY KEY,
    id_era INTEGER FOREIGN KEY REFERENCES GeoEra(id_era),
    nazwa_okres VARCHAR(50) NOT NULL
);

CREATE TABLE GeoWiek (
    id_lat INTEGER NOT NULL PRIMARY KEY,
    lata INTEGER NOT NULL
);

CREATE TABLE GeoEpoka (
    id_epoka INTEGER NOT NULL PRIMARY KEY,
    id_okres INTEGER FOREIGN KEY REFERENCES GeoOkres(id_okres),
    id_lat INTEGER FOREIGN KEY REFERENCES GeoWiek(id_lat),
    nazwa_epoka VARCHAR(50) NOT NULL
);

CREATE TABLE GeoPietro (
    id_pietro INTEGER NOT NULL PRIMARY KEY,
    id_epoka INTEGER FOREIGN KEY REFERENCES GeoEpoka(id_epoka),
    nazwa_pietro VARCHAR(50) NOT NULL
);

--Wypełnienie tabel rekordami
INSERT INTO GeoEon Values
(1, 'Fanerozoik');

INSERT INTO GeoEra Values
(1, 1, 'Paleozoik'),
(2, 1, 'Mezozoik'),
(3, 1, 'Kenozoik');

INSERT INTO GeoOkres Values
(1, 1, 'Dewon'),
(2, 1, 'Karbon'),
(3, 1, 'Perm'),
(4, 2, 'Trias'),
(5, 2, 'Jura'),
(6, 2, 'Kreda'),
(7, 3, 'TrzeciorzedPaleogen'),
(8, 3, 'TrzeciorzedNeogen'),
(9, 3, 'Czwartorzed');

INSERT INTO GeoWiek Values
(1, 395),
(2, 345),
(3, 280),
(4, 230),
(5, 195),
(6, 140),
(7, 65),
(8, 22.5),
(9, 1.8),
(10, 0.010);
```

INSERT INTO GeoEpoka Values

```
(1, 1, 1, 'Dolny'),
(2, 1, 1, 'Srodkowy'),
(3, 1, 1, 'Gorny'),
(4, 2, 2, 'Dolny'),
(5, 2, 2, 'Gorny'),
(6, 3, 3, 'Dolny'),
(7, 3, 3, 'Gorny'),
(8, 4, 4, 'Dolna'),
(9, 4, 4, 'Srodkowa'),
(10, 4, 4, 'Gorna'),
(11, 5, 5, 'Dolna'),
(12, 5, 5, 'Srodkowa'),
(13, 5, 5, 'Gorna'),
(14, 6, 6, 'Dolna'),
(15, 6, 6, 'Gorna'),
(16, 7, 7, 'Paleocen'),
(17, 7, 7, 'Eocen'),
(18, 7, 7, 'Oligocen'),
(19, 8, 8, 'Miocen'),
(20, 8, 8, 'Pliocen'),
(21, 9, 9, 'Plejstocen'),
(22, 9, 10, 'Holocen');
```

INSERT INTO GeoPietro Values

```
(1, 1, 'Lachkow'),
(2, 1, 'Prag'),
(3, 1, 'Ems'),
(4, 2, 'Eifel'),
(5, 2, 'Zywet'),
(6, 3, 'Fran'),
(7, 3, 'Famen'),
(8, 6, 'Assel'),
(9, 6, 'Sakmar'),
(10, 6, 'Artinsk'),
(11, 6, 'Kangur'),
(12, 7, 'Ufa'),
(13, 7, 'Kazan'),
(14, 7, 'Tatar'),
(15, 8, 'Ind'),
(16, 8, 'Olenek'),
(17, 9, 'Anizyk'),
(18, 9, 'Ladyn'),
(19, 10, 'Karnik'),
(20, 10, 'Noryk'),
(21, 10, 'Retyk'),
(22, 11, 'Hetang'),
(23, 11, 'Synemur'),
(24, 11, 'Pliensbach'),
(25, 11, 'Toark'),
(26, 12, 'Aalen'),
(27, 12, 'Bajos'),
(28, 12, 'Baton'),
(29, 12, 'Kelowej'),
(30, 13, 'Oksford'),
(31, 13, 'Kimeryd'),
(32, 13, 'Tyton'),
(33, 14, 'Berias'),
(34, 14, 'Walanzyn'),
(35, 14, 'Hoteryw'),
(36, 14, 'Barrem'),
(37, 14, 'Apt'),
(38, 14, 'Alb'),
(39, 15, 'Cenoman'),
(40, 15, 'Turon'),
(41, 15, 'Koniak'),
(42, 15, 'Santon'),
(43, 15, 'Kampan'),
(44, 15, 'Mastrycht'),
(45, 16, 'Dan'),
(46, 16, 'Zeland'),
(47, 16, 'Tanet'),
(48, 17, 'Iprez'),
(49, 17, 'Lutet'),
(50, 17, 'Barton'),
(51, 17, 'Priabon'),
(52, 18, 'Rupel'),
(53, 18, 'Szat'),
(54, 19, 'Akwitan'),
(55, 19, 'Burdygal'),
(56, 19, 'Lang'),
(57, 19, 'Serrawal'),
(58, 19, 'Torton'),
(59, 19, 'Mesyn'),
(60, 20, 'Zankl'),
(61, 20, 'Piacent'),
(62, 20, 'Gelas');
```

```

--Stworzenie tabeli "GeoTabela" będącej zdenormalizowaną formą stworzonych wcześniej tabel
SELECT GeoPietro.id_pietro, GeoPietro.nazwa_pietro, GeoEpoka.id_epoka,
GeoEpoka.nazwa_epoka, GeoOkres.id_okres, GeoOkres.nazwa_okres, GeoEra.id_era,
GeoEra.nazwa_era, GeoEon.id_eon, GeoEon.nazwa_eon
INTO GeoTabela
FROM GeoPietro
JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon;

--Dodanie do tabeli GeoTabela klucza głównego
ALTER TABLE GeoTabela
ADD PRIMARY KEY (id_pietro);

-----

--Utworzenie tabeli o nazwie "Dziesiec"
CREATE TABLE Dziesiec (
cyfra INTEGER NOT NULL PRIMARY KEY
);

--Wypełnienie tabeli Dziesiec rekordami
INSERT INTO Dziesiec VALUES(1);
INSERT INTO Dziesiec VALUES(2);
INSERT INTO Dziesiec VALUES(3);
INSERT INTO Dziesiec VALUES(4);
INSERT INTO Dziesiec VALUES(5);
INSERT INTO Dziesiec VALUES(6);
INSERT INTO Dziesiec VALUES(7);
INSERT INTO Dziesiec VALUES(8);
INSERT INTO Dziesiec VALUES(9);

--Stworzenie tabeli o nazwie "Milion"
CREATE TABLE Milion (
liczba INTEGER NOT NULL PRIMARY KEY
);

--Wypełnienie tabeli Milion rekordami
WITH ID(number)
AS
(
SELECT 1 AS number
UNION ALL
SELECT number + 1
FROM ID
WHERE number < 1000000
)
INSERT INTO Milion
SELECT number
FROM ID
OPTION(maxrecursion 0)

--Polecenie służące do wyświetlenia czasu wykonania zapytania
SET STATISTICS IO, TIME ON

--Zapytanie 1 (1 ZL), którego celem jest złączenie syntetycznej tablicy miliona
--wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku
--złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn:
SELECT COUNT(*) AS "1ZL"
FROM Milion
JOIN GeoTabela ON (Milion.liczba % 62) = GeoTabela.id_pietro;

```



```

--Zapytanie 2 (2 ZL), którego celem jest złączenie syntetycznej tablicy miliona
--wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez
--złączenia pięciu tabel:
SELECT COUNT(*) AS "2ZL"
FROM Milion
JOIN GeoPietro ON (Milion.liczba % 62) = GeoPietro.id_pietro
JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon

--Zapytanie 3 (3 ZG), którego celem jest złączenie syntetycznej tablicy miliona
--wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie
--jest wykonywane poprzez zagnieżdżenie skorelowane:
SELECT COUNT(*) AS "3ZG"
FROM Milion
WHERE (Milion.liczba % 62) =
(SELECT id_pietro
FROM GeoTabela
WHERE (Milion.liczba % 62) = (id_pietro));

--Zapytanie 4 (4 ZG), którego celem jest złączenie syntetycznej tablicy miliona
--wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest
--wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem
--tabel poszczególnych jednostek geochronologicznych:
SELECT COUNT(*) AS "4ZG"
FROM Milion
WHERE (Milion.liczba % 62) IN
(SELECT GeoPietro.id_pietro
FROM GeoPietro
JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon)

```


PostgreSQL:

```
1  --Stworzenie sześciu tabel oraz dodanie do nich kluczy obcych
2  CREATE TABLE GeoEon (
3  id_eon INTEGER NOT NULL PRIMARY KEY,
4  nazwa_eon VARCHAR(50) NOT NULL
5  );
6
7  CREATE TABLE GeoEra (
8  id_era INTEGER NOT NULL PRIMARY KEY,
9  id_eon INTEGER REFERENCES GeoEon(id_eon),
10 nazwa_era VARCHAR(50) NOT NULL
11 );
12
13 CREATE TABLE GeoOkres (
14 id_okres INTEGER NOT NULL PRIMARY KEY,
15 id_era INTEGER REFERENCES GeoEra(id_era),
16 nazwa_okres VARCHAR(50) NOT NULL
17 );
18
19 CREATE TABLE GeoWiek
20 (
21 id_lat INTEGER NOT NULL PRIMARY KEY,
22 lat INTEGER NOT NULL
23 );
24 CREATE TABLE GeoEpoka (
25 id_epoka INTEGER NOT NULL PRIMARY KEY,
26 id_okres INTEGER REFERENCES GeoOkres(id_okres),
27 id_lat INTEGER REFERENCES GeoWiek(id_lat),
28 nazwa_epoka VARCHAR(50) NOT NULL);
29
30 CREATE TABLE GeoPietro (
31 id_pietro INTEGER NOT NULL PRIMARY KEY,
32 id_epoka INTEGER REFERENCES GeoEpoka(id_epoka),
33 nazwa_pietro VARCHAR(50) NOT NULL
34 );
35
36
37 --Wypełnienie tabel rekordami
38 INSERT INTO GeoEon Values
39 (1, 'Fanerozoik');
40
41 INSERT INTO GeoEra Values
42 (1, 1, 'Paleozoik'),
43 (2, 1, 'Mezozoik'),
44 (3, 1, 'Kenozoik');
45
46 INSERT INTO GeoOkres Values
47 (1, 1, 'Dewon'),
48 (2, 1, 'Karbon'),
49 (3, 1, 'Perm');
```

```

50 (4, 2, 'Trias'),
51 (5, 2, 'Jura'),
52 (6, 2, 'Kreda'),
53 (7, 3, 'TrzeciorzedPaleogen'),
54 (8, 3, 'TrzeciorzedNeogen'),
55 (9, 3, 'Czwartorzed');

```

```

56
57 INSERT INTO GeoWiek Values

```

```

58 (1, 395),
59 (2, 345),
60 (3, 280),
61 (4, 230),
62 (5, 195),
63 (6, 140),
64 (7, 65),
65 (8, 22.5),
66 (9, 1.8),
67 (10, 0.010);

```

```

68
69 INSERT INTO GeoEpoka Values

```

```

70 (1, 1, 1, 'Dolny'),
71 (2, 1, 1, 'Srodkowy'),
72 (3, 1, 1, 'Gorny'),
73 (4, 2, 2, 'Dolny'),
74 (5, 2, 2, 'Gorny'),

```

```

75 (6, 3, 3, 'Dolny'),
76 (7, 3, 3, 'Gorny'),
77 (8, 4, 4, 'Dolna'),
78 (9, 4, 4, 'Srodkowa'),
79 (10, 4, 4, 'Gorna'),
80 (11, 5, 5, 'Dolna'),
81 (12, 5, 5, 'Srodkowa'),
82 (13, 5, 5, 'Gorna'),
83 (14, 6, 6, 'Dolna'),
84 (15, 6, 6, 'Gorna'),
85 (16, 7, 7, 'Paleocen'),
86 (17, 7, 7, 'Eocen'),
87 (18, 7, 7, 'Oligocen'),
88 (19, 8, 8, 'Miocen'),
89 (20, 8, 8, 'Pliocen'),
90 (21, 9, 9, 'Plejstocen'),
91 (22, 9, 10, 'Holocen');

```

```

92
93 INSERT INTO GeoPietro Values

```

```

94 (1, 1, 'Lachkow'),
95 (2, 1, 'Prag'),
96 (3, 1, 'Ems'),
97 (4, 2, 'Eifel'),
98 (5, 2, 'Zywet'),
99 (6, 3, 'Fran'),

```

100	(7, 3, 'Famen'),
101	(8, 6, 'Assel'),
102	(9, 6, 'Sakmar'),
103	(10, 6, 'Artinsk'),
104	(11, 6, 'Kangur'),
105	(12, 7, 'Ufa'),
106	(13, 7, 'Kazan'),
107	(14, 7, 'Tatar'),
108	(15, 8, 'Ind'),
109	(16, 8, 'Olenek'),
110	(17, 9, 'Anizyk'),
111	(18, 9, 'Ladyn'),
112	(19, 10, 'Karnik'),
113	(20, 10, 'Noryk'),
114	(21, 10, 'Retyk'),
115	(22, 11, 'Hetang'),
116	(23, 11, 'Synemur'),
117	(24, 11, 'Pliensbach'),
118	(25, 11, 'Toark'),
119	(26, 12, 'Aalen'),
120	(27, 12, 'Bajos'),
121	(28, 12, 'Baton'),
122	(29, 12, 'Kelowej'),
123	(30, 13, 'Oksford'),

124	(31, 13, 'Kimeryd'),
125	(32, 13, 'Tyton'),
126	(33, 14, 'Berias'),
127	(34, 14, 'Walanzyn'),
128	(35, 14, 'Hoteryw'),
129	(36, 14, 'Barrem'),
130	(37, 14, 'Apt'),
131	(38, 14, 'Alb'),
132	(39, 15, 'Cenoman'),
133	(40, 15, 'Turon'),
134	(41, 15, 'Koniak'),
135	(42, 15, 'Santon'),
136	(43, 15, 'Kampan'),
137	(44, 15, 'Mastrycht'),
138	(45, 16, 'Dan'),
139	(46, 16, 'Zeland'),
140	(47, 16, 'Tanet'),
141	(48, 17, 'Iprez'),
142	(49, 17, 'Lutet'),
143	(50, 17, 'Barton'),
144	(51, 17, 'Priabon'),
145	(52, 18, 'Rupel'),
146	(53, 18, 'Szat'),
147	(54, 19, 'Akwitan'),

```

148 (55, 19, 'Burdygal'),
149 (56, 19, 'Lang'),
150 (57, 19, 'Serrawal'),
151 (58, 19, 'Torton'),
152 (59, 19, 'Mesyn'),
153 (60, 20, 'Zankl'),
154 (61, 20, 'Piacent'),
155 (62, 20, 'Gelas');
156
157
158 --Stworzenie tabeli "GeoTabela" będącej zdenormalizowaną formą stworzonych wcześniej tabel
159 SELECT GeoPietro.id_pietro, GeoPietro.nazwa_pietro, GeoEpoka.id_epoka,
160 GeoEpoka.nazwa_epoka, GeoOkres.id_okres, GeoOkres.nazwa_okres, GeoEra.id_era,
161 GeoEra.nazwa_era, GeoEon.id_eon, GeoEon.nazwa_eon
162 INTO GeoTabela
163 FROM GeoPietro
164 JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
165 JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
166 JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
167 JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon;
168
169 --Dodanie do tabeli GeoTabela klucza głównego
170 ALTER TABLE GeoTabela
171 ADD PRIMARY KEY (id_pietro);

```

```

172
173
174 -----
175
176 --Utworzenie tabeli o nazwie "Dziesiec"
177 CREATE TABLE Dziesiec (
178 cyfra INTEGER NOT NULL PRIMARY KEY
179 );
180
181 --Wypełnienie tabeli Dziesiec rekordami
182 INSERT INTO Dziesiec VALUES(1);
183 INSERT INTO Dziesiec VALUES(2);
184 INSERT INTO Dziesiec VALUES(3);
185 INSERT INTO Dziesiec VALUES(4);
186 INSERT INTO Dziesiec VALUES(5);
187 INSERT INTO Dziesiec VALUES(6);
188 INSERT INTO Dziesiec VALUES(7);
189 INSERT INTO Dziesiec VALUES(8);
190 INSERT INTO Dziesiec VALUES(9);
191
192 --Stworzenie tabeli o nazwie "Milion"
193 CREATE TABLE Milion (
194 liczba INTEGER NOT NULL PRIMARY KEY
195 );

```

```

196
197 --Wypełnienie tabeli Milion rekordami
198 WITH RECURSIVE ID(number)
199 AS
200 (
201     SELECT 1 AS number
202     UNION ALL
203     SELECT number + 1
204     FROM ID
205     WHERE number < 1000000
206 )
207 INSERT INTO Milion
208     SELECT number
209     FROM ID
210
211
212 --Dodanie indeksów do tabeli
213 CREATE INDEX id_eon ON GeoEon(id_eon);
214 CREATE INDEX id_era ON GeoEra(id_era);
215 CREATE INDEX id_epoka ON GeoEpoka(id_epoka);
216 CREATE INDEX id_okres ON GeoOkres(id_okres);
217 CREATE INDEX id_piet ON GeoPietro(id_pietro);
218 CREATE INDEX id_lat ON GeoWiek(id_lat);
219 CREATE INDEX liczba ON Milion(liczba);

```

```

220 CREATE INDEX cyfra ON Dziesiec(cyfra);
221 CREATE INDEX id_pietro ON GeoTabela(id_pietro);
222
223 --Zapytanie 1 (1 ZL), którego celem jest złączenie syntetycznej tablicy miliona
224 --wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku
225 --złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn:
226 SELECT COUNT(*) AS "1ZL"
227 FROM Milion
228 JOIN GeoTabela ON (Milion.liczba % 62) = GeoTabela.id_pietro;
229
230 --Zapytanie 2 (2 ZL), którego celem jest złączenie syntetycznej tablicy miliona
231 --wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez
232 --złączenia pięciu tabel:
233 SELECT COUNT(*) AS "2ZL"
234 FROM Milion
235 JOIN GeoPietro ON (Milion.liczba % 62) = GeoPietro.id_pietro
236 JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
237 JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
238 JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
239 JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon
240
241 --Zapytanie 3 (3 ZG), którego celem jest złączenie syntetycznej tablicy miliona
242 --wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie
243 --jest wykonywane poprzez zagnieżdżenie skorelowane:

```

```

244 SELECT COUNT(*) AS "3ZG"
245 FROM Milion
246 WHERE (Milion.liczba % 62) =
247 (SELECT id_pietro
248 FROM GeoTabela
249 WHERE (Milion.liczba % 62) = (id_pietro));
250
251 --Zapytanie 4 (4 ZG), którego celem jest złączenie syntetycznej tablicy miliona
252 --wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest
253 --wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem
254 --tabel poszczególnych jednostek geochronologicznych:
255 SELECT COUNT(*) AS "4ZG"
256 FROM Milion
257 WHERE (Milion.liczba % 62) IN
258 (SELECT GeoPietro.id_pietro
259 FROM GeoPietro
260 JOIN GeoEpoka ON GeoEpoka.id_epoka = GeoPietro.id_epoka
261 JOIN GeoOkres ON GeoOkres.id_okres = GeoEpoka.id_okres
262 JOIN GeoEra ON GeoEra.id_era = GeoOkres.id_era
263 JOIN GeoEon ON GeoEon.id_eon = GeoEra.id_eon)

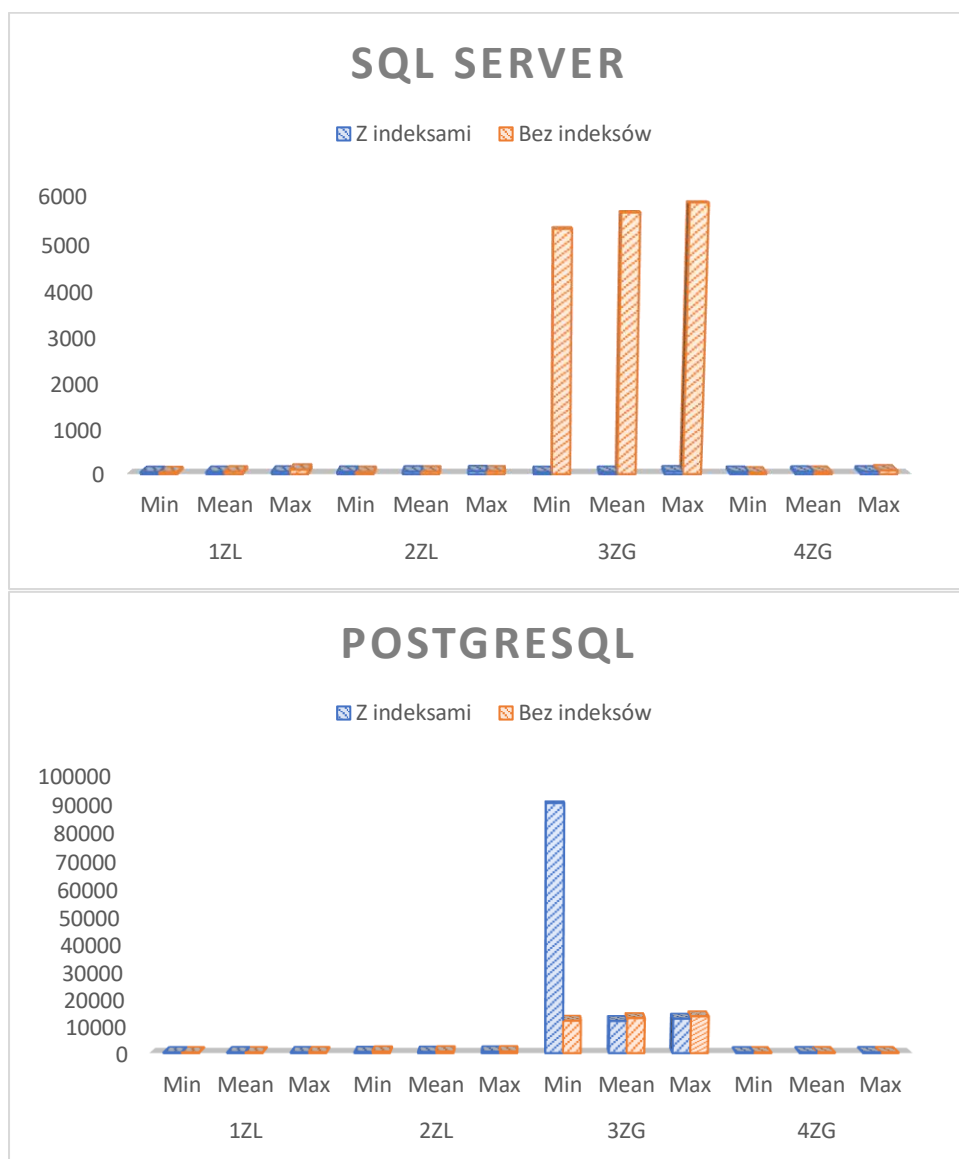
```

Wyniki testów:

Tabele przedstawiają minimalny, średni oraz maksymalny czas wykonania każdego z zapytań dla każdego z wykorzystanych systemów zarządzania bazą danych, odpowiednio z lub bez indeksów

	Z indeksami											
	1ZL			2ZL			3ZL			4ZL		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
SQL Server	44	47	54	50	56	65	44	48	58	48	52	55
PostgreSQL	241	262	273	392	420	502	90040	11962	12794	253	264	279
	Bez indeksów											
	1ZL			2ZL			3ZL			4ZL		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
SQL Server	43	56	100	48	55	59	5302	5641	5847	35	49	81
PostgreSQL	232	254	297	520	552	582	12063	13046	13704	226	260	281

Wyniki przedstawione w formie grafu:



Wnioski:

- ❖ W porównaniu do PostgreSQL, SQL Server znacznie szybciej wykonuje zapytania zarówno z indeksami, jak i bez indeksów.
- ❖ W obu użytych systemach zarządzania bazami danych zapytania z indeksami wykonują się szybciej.
- ❖ Zagnieżdżenia skorelowane są dużo wolniejsze w wykonaniu niż złączenia.
- ❖ Indeksacja w PostgreSQL wydłużyła czas wykonania zapytania 4ZG.
- ❖ W obu systemach zarządzania bazami danych zapytanie zagnieżdżone 3ZG bez indeksów wykonuje się wyraźnie wolniej od pozostałych.